53.301-298 Standard Form 298, Report Documentation Page.

AFOSR-TR-97-0416

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
95/11/22

3. REPORT TYPE AND DATES COVERED
FINAL - 94/09/01-95/08/31

4. TITLE AND SUBTITLE
Thrust-Induced Effects on a Pitching-Up Delta Wing Flow Field: Control of Stalled Wings

5. FUNDING NUMBERS
F49620-93-1-0568

6. AUTHOR(S)
L. Van Dommelen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Florida A&M University
FAMU/FSU College of Engineering
Department of Mechanical Engineering
2525 Pottsdamer St., Room 229
Tallahassee, FL. 32310

8. PERFORMING ORGANIZATION REPORT NUMBER
FMRL-TR95-2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Division of Aerospace Sciences
Bolling AFB
Washington, DC. 20332-6448

NA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Neural network procedures were explored with the objectives to control a delta wing in the non-linear domain. Results show that nerual networks are well suited to represent non-linear computed lift and pressure forces on delta wings. A model problem, the flare maneuver, suggests that neural network controllers can perform well for non-linear systems provided that suitable input and training data are provided to it.

7066a

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Introduction

The purpose of this investigation was to explore using neural networks to control a delta wing.

Previous Work

The investigation started with a literature search into past work on neural networks. The search produced work ranging from the fundamental like (Simpson, P. 1992), to control of dynamic systems (Tzirkel-Hancock and Fallside, 1991) and (Phillips and Muller- Dott, 1992), to recent work done for the control of aircraft by (Faller, Schreck and Luttges, 1993). This work was studied to help motivate and guide the direction of research.

Evaluation of Software

Research was done into software packages available for simulating neural network training. Requirements identified for a neural network simulator were versatility, user friendliness, cost, and output. Three packages were looked at, one commercial and two available free from FTP sites. The first package was Public Domain Neural Network (PDNNET) developed by Bruce Colletti. PDNNET implements a "plain vanilla" backpropagation neural network without momentum. The package was a text input-output program written in C. While the program was very easy to use, it lacked any versatility in the way of choosing activation functions, output functions, or learning procedures than the one hard coded in the program.

The commercial package looked at was Matlab with the neural simulator. The package available was a text version. The drawbacks with this package were the accessibility of the program on the work stations and the speed.

The final package was the Stuttgart Neural Network Simulator (SNNS) which is available by FTP from the University of Stuttgart in Germany. The program required some work to get it working on the IBM RISC 6000 computers at SCRI. SNNS's simulator kernel is written in C and uses a graphical user interface which runs under X11R4 or R5. The current version has nineteen available learning procedures along with several different activation and output functions available. Networks can be created and trained all within the graphical displays of the program. From this point on, most simulations discussed were created and trained using SNNS. However, Matlab was used to model the complicated of the flare maneuver.

Lift

The first investigation into the performance of neural networks was to predict the coefficient of lift when given the angle of attack of a NACA 2412 wing. The network consisted of 1 input (angle), seven hidden nodes, and one output (coefficient of lift). The network used a logistic activation function and was trained using standard backpropagation. The training data consisted of seventeen normalized angles and coefficients of lift. The network was trained for 100,000 cycles to less than .05 percent error. The network was then tested on both training data as well as new data. The network reproduced the coefficient of lift very accurately.

The same network was trained again, but this time only five hidden nodes were used. The network trained for 100,000 cycles to less than .05 percent error. Again the network reproduced the coefficient of lift curve very well, but not to the same accuracy as the first test.

The data that had been used to train the networks was obtained from John Anderson's 'Introduction to Flight'. The data was "noisy" due to the interpolation from the graph. By training the networks, the noise was eliminated and the output was a curve fit to the training data.

Angle of Attack

Next attention was turned to creating a neural network that would predict the angle of attack of a NACA 2412 wing when given the coefficient of lift. This is a singular relationship due to the occurrence of stall. The network consisted of 2 inputs (coefficient of lift and stall), two hidden layers consisting of seven and two hidden nodes respectively, and one output(angle). The stall input was used to indicate when the wing was in the stall region. The value of the stall input was either one or zero depending on if the wing was in the stall region or not.

The network used a logistic activation function and trained using standard backpropagation. The training data consisted of seventeen normalized sets of coefficient of lift, stall, and angle. The network was trained for 100,000 cycles to less than .05 percent error. The network was then tested with both new and training data.

The training and testing of the network did not do as good a job of curve fitting the data as in the previous network. The network had a particularly hard time at the onset of stall when the slope of the curve became infinite.

Lift of Slender Delta Wings

The next investigation was into the representation of the steady lift of a slender delta wing with leading edge vortices. To generate the training data, a panel method along the general lines proposed proposed by (Mangler and Smith, 1959) was developed. Using these results, a neural network was trained to predict the pressure and lift forces. It was found that the neural network represented these non-linear quantities very well.

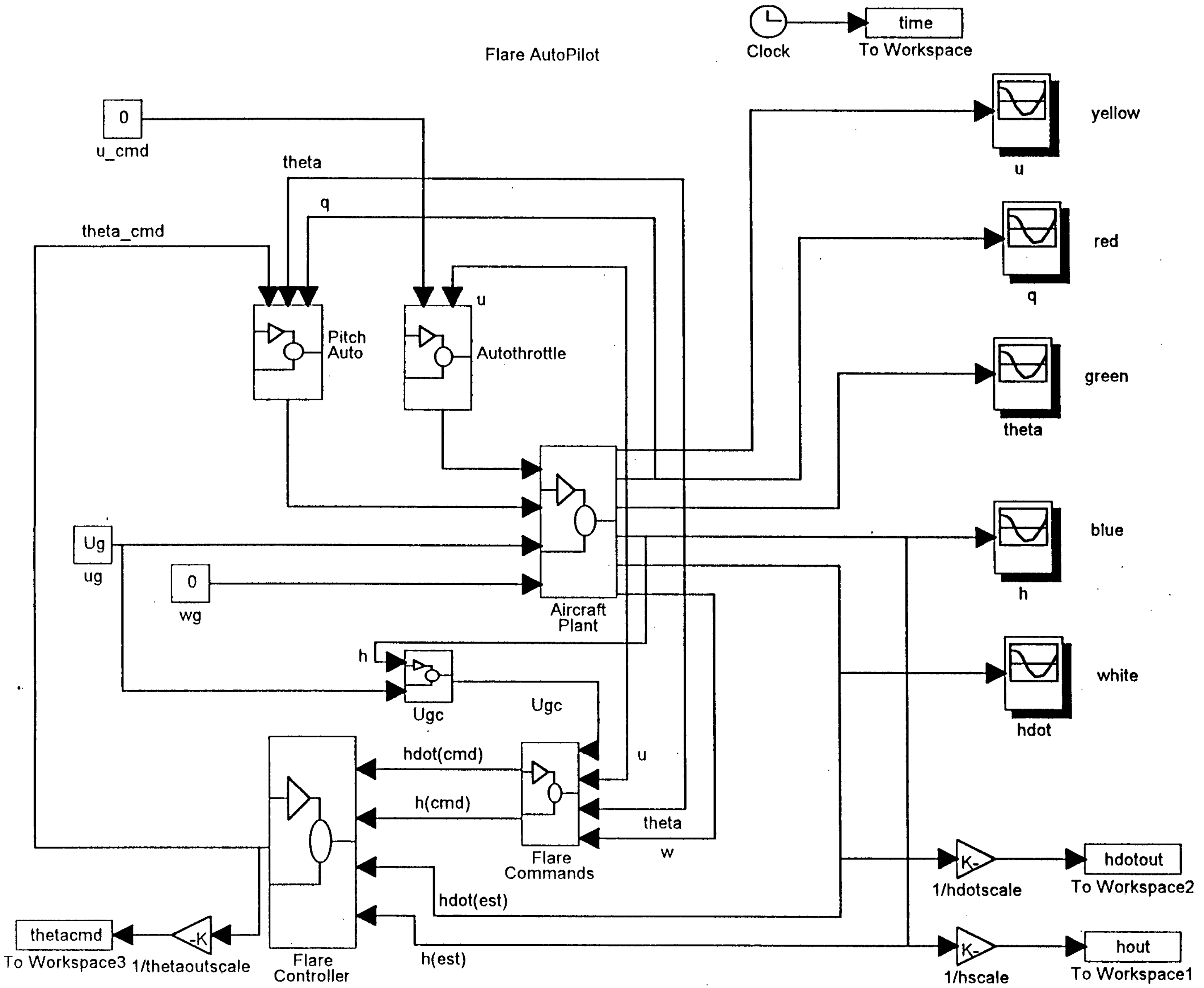
Flare Maneuver

Control of wings in the stalled regime is complicated not only by non-linearity, but also by significant history effects. In the neural network literature, the standard problem is backing up a truck. However, a model problem more closely related to our interest is that of the flare maneuver. Here too there is no direct relationship between the instantaneous flow variables and the desired reponse, but the final result, touch-down, depends on the entire previous history. It is well known that a landing that is poorly initiated is unlikely to be perfect.

Because of such considerations, a study was made of the application of neural networks to an auto-landing system. A study by did not find any advantage to a neural network compared to a standard linear controller. However, it was felt that this was likely to be due to the fact that the network was trained from the linear controller. Furthermore, an integrated history of one variable used by the linear controller was not provided as input to the neural network.

The Matlab Simulink package was used to model the complete aircraft system for both the glide and flare phases. As an example, figure 1 shows the model for the flare autopilot. From this model, data were generated that were exported to the SNNS package. All variables used by the linear controller were provided as inputs to the network, and the output of the controller was used as desired output. It was found that the modified neural network could reproduce the linear controller accurately.

In order to provide a significantly better landing system, clearly the linear controller cannot be used as measure. Hence future work should provide the desired conditions at touch-down. Such training can be performed backwards, where initially the neural network is learned to improve upon the linear response during the very last stages of the maneuver. Having learned that, the training is subsequently extended to cover earlier phases. Such procedures have already proven effective for the problem of backing up a truck, and they should apply here as well.


Flare AutoPilot
Clock
time
To Workspace
0
u_cmd
theta
q
theta_cmd
u
Pitch Auto
Autothrottle
yellow
u
red
q
green
theta
Ug
ug
0
wg
Aircraft Plant
blue
h
white
hdot
h
Ugc
Ugc
hdot(cmd)
h(cmd)
u
theta
w
Flare Commands
hdot(est)
h(est)
Flare Controller
thetacmd
To Workspace3
-K-
1/thetaoutscale
K-
1/hdotscale
hdotout
To Workspace2
K-
1/hscale
hout
To Workspace1